

# Building lexical resources for PrincPar, a large coverage parser that generates principled semantic representations

Rajen Subba\*, Barbara Di Eugenio\*, Elena Terenzi†

\*Computer Science  
University of Illinois  
Chicago, IL, USA  
{rsubba, bdieugen}@cs.uic.edu  
† Elettronica e Informazione  
Politecnico di Milano  
Milano, Italy  
elenaterenzi@hotmail.com

## Abstract

Parsing, one of the more successful areas of Natural Language Processing, has mostly been concerned with syntactic structure. Though uncovering the syntactic structure of sentences is very important, in many applications a meaning representation for the input must be derived as well. We report on PrincPar, a parser that builds full meaning representations. It integrates LCFLEX, a robust parser, with a lexicon and ontology derived from two lexical resources, VerbNet and CoreLex, that represent the semantics of verbs and nouns respectively. We show that these two different lexical resources that focus on verbs and nouns can be successfully integrated. We report parsing results on a corpus of instructional text and assess the coverage of those lexical resources. Our evaluation metric is the number of verb frames that are assigned a correct semantics: 72.2% verb frames are assigned a perfect semantics, and another 10.9% are assigned a partially correct semantics. Our ultimate goal is to develop a (semi)automatic method to derive domain knowledge from instructional text, in the form of linguistically motivated action schemes.

## 1. Introduction

Current parsers reach accuracies between 86% and 90%, as measured by different types of precision and recall ((Charniak, 2000), (Collins, 2003)). These results though only concern syntactic structure. While uncovering syntactic structure is certainly valuable per se, in many applications a meaning representation for the input must be derived as well. There have been results on head-dependency recovery rates reported by several parsers, such as (Carroll et al., 1998) and (Collins, 2003). Although head-dependencies are related to semantic relations, they fall far from full meaning representations.

The goal of building meaning representations is of course not new. Unfortunately, many symbolic parsers from the '70s and '80s were brittle and non robust. Alternatively, semantic parsers which don't build a syntactic structure but only a semantic one work well but only in restricted domains.

Certainly one of the bottlenecks towards building a large coverage parser that derives semantic representations is access to large coverage semantic resources. Resources, such as VerbNet (Kipper et al., 2000) and FrameNet (Baker et al., 2003), have only recently started to become available. WordNet (Fellbaum, 1998) has been available for much longer than VerbNet and FrameNet. However, as much as WordNet has greatly affected computational work and is used in an extremely large number of projects, it does not attempt at providing an explicit semantics for the words it includes.

In this paper, we discuss the integration of LCFLEX (Rosé and Lavie, 2000) with VerbNet (Kipper et al., 2000) and CoreLex (Buitelaar, 1998) to build PrincPar. We demonstrate that indeed these three components together

can achieve very good parsing results. Our evaluation metric is the number of verb frames that are assigned a correct semantics: 72.2% verb frames are assigned a perfect semantics, and another 10.9% are assigned a partially correct semantics. Our testing data is instructional text. The corpus is about 9MB in size and is made up entirely of written English instructions.<sup>1</sup>

We are interested in accounting for examples such as the following:

(1a) *Wipe the fingerprints from the counter.*

(1b) *Wipe the counter.*

(2a) *Remove the groceries from the bag.*

(2b) *Remove the bag.*

As the effect of the two actions (1a) and (2a), it is inferred that the specified location (*counter* in (1a), *bag* in (2a)) has been "emptied" of the theme (*fingerprints* in (1a), *groceries* in (2a)). Thus, a system could map both verbs *wipe* and *remove* onto the same action scheme. However, the apparently equivalent transformations from (1a) to (1b) and from (2a) to (2b) show otherwise. (1b) describes the same action as (1a), however (2b) cannot have the same meaning as (2a). Such linguistic phenomena are captured in (Levin and Rappaport Hovav, 1992) by defining classes of verbs according to the ability or inability of a verb to occur in pairs of syntactic frames that preserve meaning.

We chose to base our lexicon and ontology on VerbNet

<sup>1</sup>The two largest components are home repair manuals (5Mb) and cooking recipes (1.7Mb). It was collected opportunistically off the internet and from other sources, and originally assembled at the Information Technology Research Institute, University of Brighton.

```

( :morph      paper
  :syntax     (*or* ((cat n) (root paper) (agr 3s) (countable mass)(semtag art1))
              ((cat vlex) (root paper) (vform bare) (aroot over)(subcat (*or* part-np np))
                           (semtag butter-9.9-Transitive_-Destination_Object-Agent_V_Destination))
              ((cat vlex) (root paper) (vform bare) (aroot over)(subcat np-pp) (proot with)
                           (semtag butter-9.9-NP-PP_-Theme-PP-Agent_V_Destination_Prep-with-_Theme)))
  :semantics  (*or* (butter-9.9-Transitive_-Destination_Object-Agent_V_Destination
                    (<butter-9.9-Transitive_-Destination_Object-Agent_V_Destination> (subj agt)(obj dest)))
              (butter-9.9-NP-PP_-Theme-PP-Agent_V_Destination_Prep-with-_Theme
                (<butter-9.9-NP-PP_-Theme-PP-Agent_V_Destination_Prep-with-_Theme> (subj agt)(obj dest)(pred thm)))
              (art1 (<art1>))
              (art2 (<art2>))))

( :morph      remove
  :syntax     (*or* ((cat n) (root remove) (agr 3s) (countable count))
              ((cat vlex) (root remove) (vform bare) (features vveryvingpast)(proot to) (subcat np-pp)
                           (semtag banish-10.2-NP-PP_-Destination-PP-Agent_V_Theme_Prep-to-_Destination))
              ((cat vlex) (root remove) (vform bare) (features vveryvingpast)(subcat np)
                           (semtag (*or* banish-10.2-Basic_TransitiveAgent_V_Theme remove-10.1-Basic_TransitiveAgent_V_Theme)))
              ((cat vlex) (root remove) (vform bare) (features vveryvingpast) (proot from) (subcat np-pp)
                           (semtag (*or* banish-10.2-NP-PP_-Source-PP-Agent_V_Theme_Prep[+src]_Source
                                   remove-10.1-NP-PP_-Source-PP-Agent_V_Theme_Prep[+src]_Source)))
              .....
              ((cat vlex) (root remove) (vform bare) (features vveryvingpast)(proot (*or* from to)) (subcat np-pp-pp)))
  :semantics  (*or* (banish-10.2-Basic_TransitiveAgent_V_Theme
                    (<banish-10.2-Basic_TransitiveAgent_V_Theme> (subj agt)(obj thm)))
              .....
              (remove-10.1-Basic_TransitiveAgent_V_Theme
                (<remove-10.1-Basic_TransitiveAgent_V_Theme> (subj agt) (obj thm)))
              (remove-10.1-NP-PP_-Source-PP-Agent_V_Theme_Prep[+src]_Source
                (<remove-10.1-NP-PP_-Source-PP-Agent_V_Theme_Prep[+src]_Source> (subj agt)(obj thm)(pred src))))

```

Figure 1: The entries for *paper* and *remove* in our lexicon

(Kipper et al., 2000), that operationalizes Levin’s work and accounts for about 4962 distinct verbs classified into 237 main classes. Moreover, given VerbNet’s strong syntactic components, it can be easily coupled with a parser and used to automatically generate a semantically annotated corpus. Of course, when building a representation for a sentence, we need semantics for nouns as well. We found CoreLex (Buitelaar, 1998) appropriate for our needs. CoreLex is based on the theory of the generative lexicon (Pustejovsky, 1991), and provides a meaning representation for nouns compatible with that for verbs in VerbNet.

The contribution of our work is to demonstrate that a meaning representation based on decompositional lexical semantics can be derived efficiently and effectively, using the most current resources in the area.

We also show that two different lexical resources that focus on verbs and nouns can be successfully integrated. Further, our work constitutes an assessment of the coverage of those large scale semantic resources. While the field will certainly benefit from their availability, their coverage on real world data is not known. For example, in our test set (200 sentences randomly selected from our corpus) there are 157 distinct verbs. Although VerbNet covers 4962 verbs, it does not cover 19, i.e. 12%, of our verbs.

In the following, we describe our lexicon and ontology in Secs. 2 and 3, discuss the semantic representation that the parser produces in Sec. 4, present the evaluation in Sec. 5, and related work and discussion in Sec. 6.

## 2. Lexicon

We chose LCFlex (Rosé and Lavie, 2000), a robust left-corner parser, as the core of PrincPar, because LCFLEX can return portions of analysis when faced with ungrammaticalities or unknown words or structures (the latter is likely in a large corpus). We modified and augmented LCFLEX’s existing lexicon, based on COMLEX (Grishman et al., 1994). To illustrate our work, we will refer to one of our test sentences,

- (3) Before you can paper the wall, you must remove all existing fungal spores from the wall.

Figure 1 shows the lexical entries for *paper* that can be both a noun (*n*) or a verb (*vlex*), and for the verb *remove* that belongs to more than one verb class in VerbNet. The format of the lexicon comes from COMLEX, but the *:semantics* field was originally empty. For the verb, different subcategorization frames are listed under *subcat*: the verb can have as argument just an np, or an np and a pp, or an np and an adverbial phrase. In each entry, each part of speech (POS) category (e.g., (*cat vlex*)) is associated to a *semtag*, an index that links the POS category to the corresponding semantic representation. *<butter-9.9-Transitive\_-Destination\_Object-Agent\_V\_Destination>*, *<art1>* and *<art2>* are entries in our ontology. Before discussing the ontology, we need to discuss VerbNet and CoreLex formalisms.

Figure 2 shows the VerbNet class *butter-9.9* to which the verb *paper* belongs and the class *remove-10.1*, of which the

CLASS: butter-9.9

PARENT: -

MEMBERS: asphalt, bait, blanket, blindfold, board, bread, brick, bridle, bronze, butter, carpet, caulk, chrome ...

THEMATIC ROLES: Agent Theme Destination

SELECTIONAL RESTRICTIONS: Agent[+animate] Theme[+concrete] Destination[+location -region]

FRAMES:

|                                 |                                      |  |
|---------------------------------|--------------------------------------|--|
| Transitive (Destination Object) | Agent V Destination                  | cause(Agent, E) $\wedge$ motion(during(E), Theme) $\wedge$ $\neg$ location(start(E), Theme, Destination) $\wedge$ location(end(E), Theme, Destination) |
| Transitive (+ Theme PP)         | Agent V Destination Prep[with] Theme | cause(Agent, E) $\wedge$ motion(during(E), Theme) $\wedge$ $\neg$ location(start(E), Theme, Destination) $\wedge$ location(end(E), Theme, Destination) |

CLASS: remove-10.1

PARENT: -

MEMBERS: abstract, cull, delete, disgorge, dislodge, disengage, draw, eject, eliminate, eradicate, remove ...

THEMATIC ROLES: Agent Theme Source

SELECTIONAL RESTRICTIONS: Agent[+int\_control OR +organization] Theme[] Source[+location]

FRAMES:

|                          |                                 |  |
|--------------------------|---------------------------------|--|
| Transitive               | Agent V Theme                   | cause(Agent, E) $\wedge$ $\neg$ location(start(E), Theme, ?Source) $\wedge$ location(end(E), Theme, ?Source) |
| Transitive (+ Source PP) | Agent V Theme Prep[+src] Source | cause(Agent, E) $\wedge$ $\neg$ location(start(E), Theme, Source) $\wedge$ location(end(E), Theme, Source)   |

Figure 2: The classes butter-9.9 and remove-10.1 from VerbNet

verb *remove* is a member (*remove* is also a member of the class *banish-10.2*, which is not shown in Figure 2 due to the lack of space.). All verbs that can undergo the same syntactic alternations belong to the same class. A class includes a list of parent classes, empty in this case (verb classes are arranged in a hierarchy), its thematic roles and selectional restrictions on these. Then, it specifies all the frames associated with that class, and provides a meaning representation for each frame. Each frame is labeled with its name, and consists of the syntactic frame itself (e.g., *Agent V Theme Prep Destination*), and its semantic interpretation. *Agent*, *Theme* and *Destination* are three of various thematic roles VerbNet uses, *V* is for verb. For the verb class *butter-9.9*, the first frame is a basic transitive frame without any modifiers, whereas the second one is a transitive frame with a prepositional phrase that is realized as the theme. Restrictions are placed on the type of preposition that the frame can accept, which in this case is the preposition *with*.<sup>2</sup> Also, selectional restrictions on arguments enforce semantic constraints (like intentionality restrictions for agent roles) on concepts that can fill various thematic roles, associated with the class.

The semantics portion of a lexical entry links the deep syntactic roles built by the parser to the thematic roles in the verb class. LCFLEX assigns eight different deep syntactic roles. These roles include *subj* (subject), *obj* (object), *iobj* (indirect object), *pred* (descriptive predicate), *modifier* (adjunct modifiers) and *comp* (a clausal complement). In Figure 1, the following mappings are specified under *paper*: subject to agent (*subj agt*) and object to destination (*obj dest*) for the first frame, and additionally, *pred* to theme (*pred thm*) for the second frame .

<sup>2</sup>The type of restrictions placed on prepositions by VerbNet can be specific, as in the case above, or more general as in *src* (source) or *dest-dir* (destination-direction), which refer to classes of prepositions.

As regards nouns, CoreLex defines basic types such as *art* (artifact) or *com* (communication). Nouns are characterized by bundles of basic types. Nouns that share the same bundle are grouped in the same Systematic Polysemous Class (SPC). The resulting 126 SPCs cover about 40,000 nouns. We found that these CoreLex classes were compatible with the selectional restrictions imposed on arguments by VerbNet. However, since selectional restrictions are not defined anywhere in VerbNet, we had to decide how to map them to CoreLex types. Take for instance the restriction [*+int\_control*] that stands for intentional control. We believe that animate entities satisfy this restriction. The restriction [*+int\_control*] can then be expressed by the CoreLex basic types *anm* (animal) or *hum* (human) that resemble the *animate* property.

### 3. Ontology

VerbNet classes and CoreLex SPCs are realized as entities in our ontology. Figure 3 shows the entries for *butter-9.9*, *butter-9.9-Transitive-Destination-Object-Agent-V-Destination* and *art1*. The field *:isa* is used for inheritance purposes. Every type for a main verb class has the same parent type *<verb>*. Verb subclasses inherit the features of their parents. Each frame for a given class or a subclass is a type as well, whose parent is the class or subclass it belongs to. The SPC type *art1* is used to define an *artifact*. Since an *artifact* is a *concrete entity*, a selectional restriction placed by VerbNet, it becomes a subclass of the type *<concr-ent>*, which specifies a concrete entity in our ontology. Variables for the types are declared in *:vars*. The variables for the verb class *butter-9.9* are *agt* (agent), *thm* (theme) and *dest* (destination) which are also the variables for *butter-9.9-Transitive-Destination-Object-Agent-V-Destination* due to inheritance. SPC types do not have any variables.

The *:spec* field is the basis for building the semantic representation while parsing. The subfields of *:spec* on verb

```

(:type <butter-9.9>
:isa (<verb>)
:vars (agt thm dest)
:spec ((agent <animate> agt)
(destination <loc> dest)
(theme <concr-ent> thm)
))

(:type <butter-9.9-Transitive_-Destination_
Object-Agent_V_Destination>
:isa (<butter-9.9>)
:vars nil
:spec (
(event <>
(<event0>
(<not_located_in> dest thm)
(<in_motion> thm)
(<located_in> dest thm)
nil
agt))
))

(:type <art1>
:isa (<concr-ent>)
:spec ((artifact +)
))

```

Figure 3: Two entries in our ontology

classes are structured as *(name type-check arg)*. *arg* can be either a variable or a complex argument built with one or more functions (see *event*). *type-check* is a type constraint *arg* must satisfy to be included in the final representation. The *:spec* field for SPC types is less complex. We only embed the corresponding features from CoreLex, such as *artifact*, *state*, *animal*, into the *:spec* field. For the SPC type *art1*, we simply add the feature (*artifact +*) in its *:spec* field.

An event is bound to the action that the verb describes. In our ontology, *event* has a quintuplet structure: the predicates *start(E)*, *during(E)*, *end(E)*, *result(E)* and *cause(E)* decompose the event into four stages and the cause of the event. For *butter-9.9*, *<not\_located\_in>*, *<in\_motion>* and *<located\_in>* express the semantics of the verb at the beginning, during and the culmination of the event respectively.<sup>3</sup>

Both the lexicon and ontology were semi-automatically built. In order to build the lexicon, each entry in LCFLEX’s lexicon was parsed to check for its part of speech, subcategorization value and its preposition values. Based on these values, the semtag values for each of the lexical entry was inserted automatically. For verbs, our system first checked if it was covered by VerbNet with the given subcategorization value in the lexical entry. If the verb for that particular subcategorization was found, then its corresponding semtag value was inserted. Assigning the semtag values for nouns was a much simpler task. Once the CoreLex class to which the noun belongs to was established, the semtag value for

<sup>3</sup>Note that *<not\_located\_in>*, *<in\_motion>* and *<located\_in>* correspond to the semantic predicates *¬location(start(E), Theme, Source)*, *motion(during(E), Destination)* and *location(end(E), Theme, Source)*, (see Figure 2).

that particular class was inserted into the lexical entry. The ontology was built automatically by traversing through all the VerbNet classes (main classes and subclasses) and their syntactic frames.

The manual part of the semi-automatic generation of the lexicon and ontology involved the mapping of the VerbNet selectional restrictions to CoreLex classes in the ontology and the mapping of VerbNet alternations (frames) to the subcategorization values of COMLEX for the lexicon. Our lexicon includes 3547 verbs from the LCFLEX lexicon classified under 237 verb classes. It also includes 17492 nouns that were in LCFLEX grouped under 126 SPCs.

#### 4. The parser at work

We illustrate here the semantic representation that is built by PrincPar. Due to space limitations, we show a simplified representation only (we omit the surface syntactic structure from the parsed tree).

```

((ROOT PAPER)
(*SEM* ((VERBCLASS VNCLASS BUTTER-9.9)
(EVENT0
((START (!LOCATION (NIL WALL)))
(DURING (MOTION (NIL)))
(END (LOCATION (NIL WALL)))
(CAUSE (YOU))))
(AGENT (YOU))
(DESTINATION (WALL))))
(SUBJ
((*SEM* ((PERSON +) (INTL-CTRL-ENT +)
(CONCRETE-ENT +) (ANIMATE +)
(ROOT HUM4))) (ROOT YOU)))
(OBJ *
((*SEM ((ARTIFACT +) (CONCRETE-ENT +) (ROOT ART1)))
(ROOT WALL))))

```

Figure 4: Parser output for *paper*

Consider Example (3), *Before you can paper the wall, you must remove all existing fungal spores from the wall..* The sentence contains two verbs, *paper* and *remove* with a basic transitive frame and a transitive frame with a prepositional modifier that denotes the source respectively. When parsed, the semantic representation in Figure 4 is generated for the syntactic frame in which the verb *paper* appears. Figure 5 shows the corresponding semantic tree.

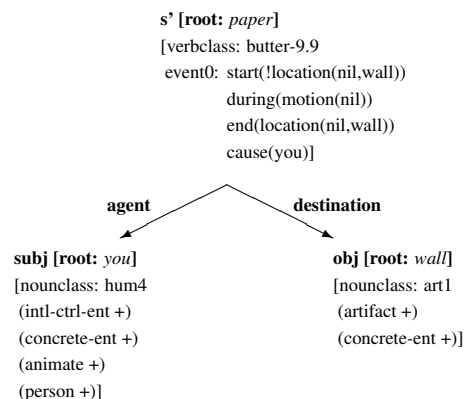


Figure 5: Semantic tree for *paper*

The topmost section represents verb semantics in terms of events (in this case there is only one event, event0). In Figure 5, the frame *Agent V Destination*, which does not in-

clude *Theme*,<sup>4</sup> is parsed. The verb class for which the verb is being parsed for is also captured in the representation.

The *subject* (*you*) and *object* (*wall*) are assigned the thematic roles *Agent* and *Destination* respectively. Semantic representations for the arguments to the verb, namely *you* and *wall* are also produced. They include the features defined in CoreLex such as (*intl-ctrl-entity +*) and (*artifact +*). The CoreLex classes they belong to are also captured (e.g. *hum4* and *art1*).

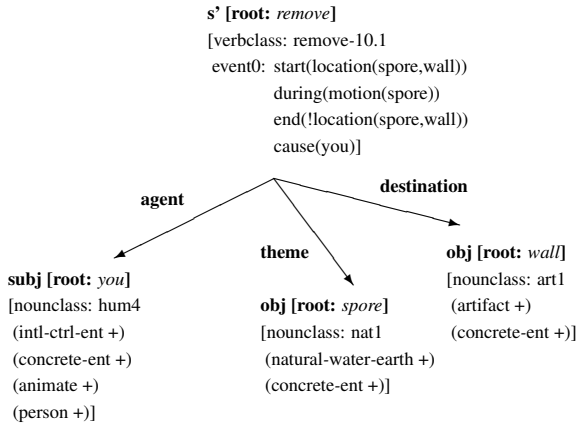


Figure 6: Semantic tree for *remove*

The semantic tree for the syntactic frame in which *remove* occurs in is shown in Figure 6<sup>5</sup>. In this case, the verb *remove* is parsed for the verb class *remove-10.1*, but not *banish-10.2* since *spores* does not satisfy the selectional restriction of being *animate*, which is required by *banish-10.2*. The frame being parsed is that of *Agent V Theme [+src] Source*. *remove-10.1* also uses the semantic predicate *location*. *spore* and *wall* are passed as the arguments to the semantic predicate *location* and thus realized as *location(spore,wall)*. Besides the *subj* and *obj* syntactic roles (that are mapped to the thematic roles *Agent* and *Theme* respectively), you will also find the syntactic role *pred*, which is mapped to the thematic role *Source*.

## 5. Evaluation and Discussion

We evaluated PrincPar on a test set taken from the home repair portion of the instructional corpus. We randomly collected 200 sentences that contained one or more action verbs.<sup>6</sup> They were parsed unmodified so as to evaluate the parser on real world data.<sup>7</sup> Since each sentence may contain more than one verb, it does not make sense to evaluate

<sup>4</sup>*nil* is used when an argument for a semantic predicate is not realized in the sentence (frame), see Figure 5.

<sup>5</sup>Due to the lack of space, we do not show the parsed semantic representation for *remove*.

<sup>6</sup>Syntactically, these 200 sentences contain a main clause plus a number of adjunct clauses. The average length of the sentences was 15.7 words.

<sup>7</sup>In an earlier evaluation (Terenzi and Di Eugenio, 2003), we reported parse results of up to 96% complete parses and 4% partial parses on a test set of 151 sentences. However, the test set in (Terenzi and Di Eugenio, 2003) was chosen so as to include one of the 109 verbs that we had in our lexicon at the time. Moreover,

whether a sentence is fully or partially parsed. Hence, our evaluation is based on the semantic representation of each verb frame in the sentence. 200 sentences may appear as a small test set. These small test sets are due to the evaluation being done manually.

|       | tokens<br>(occurrences) | entry<br>in VN | entry in VN<br>+ verb frame |
|-------|-------------------------|----------------|-----------------------------|
| verbs | 400<br>(*157)           | 344<br>(*138)  | 274                         |

Table 1: VerbNet Coverage Result (VN = VerbNet, \* refers to # of distinct verb types)

Table 1 reports our results on the coverage of VerbNet for our test data from the home-repair corpus. Our test sentences contained 157 distinct verb types for a total of 400 tokens. 138 out of 157 distinct verb types (for a total of 344 tokens) are in VerbNet and hence in our lexicon. 19 verb types are not in VerbNet. For example, *apply* which occurs 16 times in our test set is not covered by VerbNet. Of the 138 distinct verb types that did appear in the test sentences and are in VerbNet, 70 of them appeared in verb frames that have not been covered by VerbNet. For example, the entry for the verb *shop* is in VerbNet, but not the intransitive frame with just a PP attachment that specifies the theme (as in *shop for wallpaper*). In the end, we expected PrincPar to parse 274 verb frames given the lexical semantic resources available.

| complete<br>parse | partial<br>parse | wrong/missing<br>parse | total |
|-------------------|------------------|------------------------|-------|
| 198               | 30               | 46                     | 274   |

Table 2: Parsing Results

We evaluate PrincPar on the basis of whether the full semantics of a verb frame, given the lexical resources, is parsed or not. Table 2 reports our parse results. A complete parse means that the full semantic representation with all the arguments corresponding to the verb frame in VerbNet is built with every syntactic role mapped to the correct thematic role.<sup>8</sup> A partial parse means that the semantic representation of the verb frame is missing some arguments in the parsed tree.

We were able to completely parse 198 (72.2%) verb frames and partially parse 30 (10.9%) of them, for a total of 228 verb frames at least partially parsed (83.1% of the total). For the remaining 46 verb frames, LCFLEX either produced incorrect parses or failed to produce any semantic representation. LCFLEX was unable to parse progressive verbs such as *hanging* when they occur in adverbial clauses like *when hanging the frame* and in nominals like *the procedure for hanging the strips is the same*.

The remaining 126 verb frames that could not be parsed by PrincPar for lack of lexical resources provide an estimate of VerbNet’s coverage of verb semantics in the home-repair

that evaluation was based on modified sentences with only a single main verb in the clause.

<sup>8</sup>This entails that the selectional restrictions placed on the thematic roles is satisfied.

corpus. As the number of verb types and verb frames covered by VerbNet grows, we expect to be able to parse more sentences.

Reporting results as we do in Table 1 without comparing them to an appropriate baseline may appear not too telling. However, it is not clear what a suitable baseline would be in our case, since results on producing full meaning representations are just starting to appear, and so far they are not directly comparable. To our knowledge, there has not been any other attempt at using VerbNet verb semantics to build a large-coverage semantic parser. (Neville and Kipper, 2004) discusses how VerbNet has been coupled with the XTAG formalism, but no parsing results are available. Other previous work on building semantic representation such as (Gildea and Jurafsky, 2002) have focused only on the annotation of semantic roles. They report an accuracy of 82.1% using pre-segmented data and 65% precision when automatically identifying the segments and then labeling them. Our work goes beyond semantic role assignment.

In fact, the only other piece of work that we know of and that is directly related to ours is (Shi and Mihalcea, 2004). They generate full semantics for a sentence as well. However, they only report results for semantic role assignment. Their parser achieves an accuracy of 74.5% for role assignment. To provide a somewhat reasonable comparison (a direct comparison cannot be made due to the difference in data and semantic roles assigned), we evaluated PrincPar in terms of semantic role assignment as well, obtaining an accuracy of 75.1%. Importantly, unlike (Shi and Mihalcea, 2004), whose testing corpus and lexical resource are the same (i.e. FrameNet), our corpus is completely divorced from our lexical resources.

## 6. Conclusions and future work

We have shown that two rich lexicons such as VerbNet and CoreLex can be successfully integrated. We are eagerly expecting a new release of VerbNet and possibly a revised version of LCFLEX to improve our parsing results. We also intend to evaluate PrincPar on *PropBank* (Palmer et al., 2005), that recently became available via the LDC. PropBank adds verb frame semantics as defined in VerbNet to the annotated syntactic trees of the Penn TreeBank (Marcus et al., 1993). Comparing the semantics that PrincPar builds to an independently devised semantics will make our evaluation stronger. We are now poised to systematically run the parser on the full home repair portion of the corpus. Our ultimate goal is to develop a (semi)automatic method to derive domain knowledge from instructional text. We have conducted some small scale experiments on learning relations between verb classes based on their semantic information and rhetorical relations (Subba et al., 2006).

## Acknowledgments

This work is supported by award IIS-0133123 from the National Science Foundation (NSF), and additionally by awards ALT-0536968 from NSF and N000140010640 from ONR. Thanks to C.P. Rosé for LCFLEX, M. Palmer and K. Kipper for VerbNet, and C. Buitelaar for CoreLex.

## 7. References

- Baker, C. F., Fillmore, C. J., and Cronin, B. 2003. The Structure of the Framenet Database. *International Journal of Lexicography*, 16(3):281–296.
- Buitelaar, P. 1998. *CoreLex: Systematic Polysemy and Under-specification*. Ph.D. thesis, Computer Science, Brandeis University, February.
- Carroll, J., Briscoe, E., and Sanfilippo, A. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, Granada, Spain*:447–454.
- Charniak, E. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL-2000*.
- Collins, M. 2003. Head-driven statistical methods for natural language parsing. *Computational Linguistics*, 29(4).
- Fellbaum, F. 1998. *WordNet: An Electronic Lexical DataBase*. MIT Press, Cambridge, MA.
- Gildea, D. and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3).
- Grishman, R., Macleod, C., and Meyers, A. 1994. COMLEX syntax: Building a computational lexicon. In *COLING 94, Proceedings of the 15th International Conference on Computational Linguistics*, pages 472–477.
- Kipper, K., Dang, H.T., and Palmer, M. 2000. Class-based construction of a verb lexicon. In *AAAI-2000, Proceedings of the Seventeenth National Conference on Artificial Intelligence*.
- Levin, B. and Hovav, M.R. 1992. Wiping the slate clean: a lexical semantic exploration. In Beth Levin and Steven Pinker, editors, *Lexical and Conceptual Semantics, Special Issue of Cognition: International Journal of Cognitive Science*. Blackwell Publishers.
- Marcus, M., Santorini, B., and Marcinkiewicz, M. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Neville, R. and Kipper, K. 2004. Assigning Xtag Trees to VerbNet. In *Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+7)*.
- Palmer, M., Gildea, D. and Kingsbury, P. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–105.
- Pustejovsky, J. 1991. The generative lexicon. *Computational Linguistics*, 17(4):409–441.
- Rosé, C.P. and Lavie, A. 2000. Balancing robustness and efficiency in unification-augmented context-free parsers for large practical applications. In Jean-Claude Junqua and Gertjan van Noord, editors, *Robustness in Language and Speech Technology*. Kluwer Academic Press.
- Shi, L. and Mihalcea, R. 2004. Semantic parsing using framenet and wordnet. In *Proceedings of the Human Language Technology Conference (HLT/NAACL 2004)*. (Short Paper).
- Subba, R., Di Eugenio, B. and Kim, S.N. 2006. Learning FOL Rules based on Rich Verb Semantic Representations to automatically label Rhetorical Relations. In *EACL 2006, Workshop on Learning Structured Information in Natural Language Applications*.
- Terenzi, E. and Di Eugenio, B. 2003. Building lexical semantic representations for natural language instructions. In *HLT-NAACL03, 2003 Human Language Technology Conference*. (Short Paper).